AD-A017 935

CAPSULE VIEW OF FUTURE DEVELOPMENTS ON LARGE SCALE NETWORK AND NETWORK-RELATED PROBLEMS

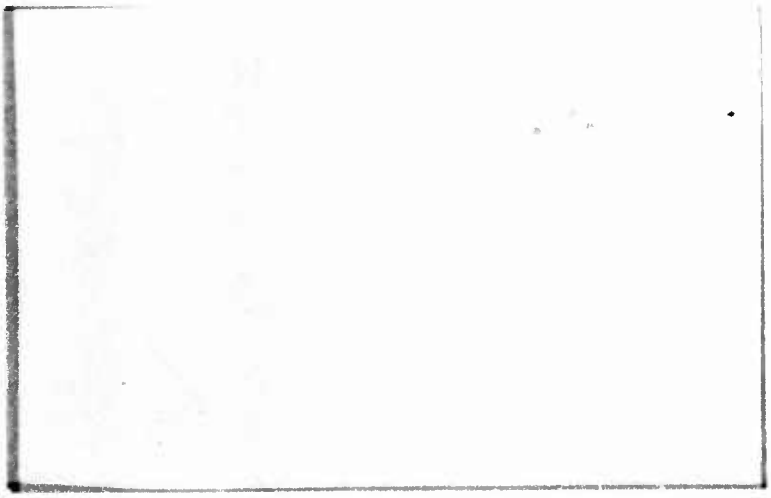
Fred Glover, et al

Texas University at Austin

Prepared for:

Office of Naval Research

June 1975



CENTER FOR CYBERNETIC STUDIES

The University of Texas
Austin, Texas 78712

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Center for Cybernetic Studies
The University of Texas

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

Capsule View of Future Developments On Large Scale Network and Network-Related Problems

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

Fred Glover
Darwin Klingman

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1975	54	

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
N00123-74-C-2272	Center for Cybernetic Studies Research Report CCS 238
b. PROJECT NO. NR 047-021	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Office of Naval Research (Code 434) Washington, D.C.

13. ABSTRACT

The primary purpose of this paper is to identify recent major accomplishments and prophesy (suggest) future trends in the solution, modeling, and human engineering aspects of network and network-related problems. In particular, the suggested future trends consist of an integrated sequence of modeling, theoretical and computational developments which may allow mathematical programming to attain its envisioned goal of being a major vehicle for solving pressing industrial and world problems.

Network and network-related problems include such mathematical problems as assignment, transportation, transshipment, multi-commodity networks, generalized networks, set covering and partitioning problems, matroid problems, plant location problems, fixed charge networks, constrained networks, general linear programs with GUB and network substructure, constrained spanning tree problems, traveling salesman problems, and combinations of these. In fact, it has recently been shown that any 0-1 integer programming problems are equivalent to 0-1 generalized network problems.

Network-related problems that arise "naturally" (i.e., those whose formulations make conspicuous reference to a network structure) cover a diverse spectrum of practical applications. These applications include problems in telecommunication, microdata set merging, cash flow, multi-national currency exchange, manpower planning, waste disposal, water resource management, aircraft refueling, modular design, ledger and account balance updating, cutting stock, structural and plastic problems of civil engineering, machine loading, aircraft scheduling, off-shore oil well location and drilling, and production and distribution scheduling, to mention only a few.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Unclassified

Security Classification

A-31408

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Networks Integer programming Linear programming Generalized networks Large scale optimization						

Research Report CCS 238

CAPSULE VIEW OF FUTURE DEVELOPMENTS ON LARGE SCALE NETWORK AND NETWORK-RELATED PROBLEMS

by

Fred Glover*

Darwin Klingman**

June 1975

*Professor of Management Science, University of Colorado, Boulder, Colorado 80302

**Professor of Operations Research and Computer Sciences and Director of Computer Sciences Research for the Center for Cybernetic Studies, University of Texas, Austin, Texas 78712

This research was partly supported by the Navy Personnel Research and Development Laboratory Contract N00123-74-C-2272 with the Center of Cybernetic Studies, The University of Texas and the Alexander-von-Humboldt-Stiftung in Bonn, Germany. Reproduction in whole or in part is permitted for any purpose of the United States Government.

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director
Business-Economics Building, 512
The University of Texas
Austin, Texas 78712
(512) 471-1821

ABSTRACT

The primary purpose of this paper is to identify recent major accomplishments and prophesy (suggest) future trends in the solution, modeling, and human engineering aspects of network and network-related problems. In particular, the suggested future trends consist of an integrated sequence of modeling, theoretical and computational developments which may allow mathematical programming to attain its envisioned goal of being a major vehicle for solving pressing industrial and world problems.

Network and network-related problems include such mathematical problems as assignment, transportation, transshipment, multi-commodity networks, generalized networks, set covering and partitioning problems, matroid problems, plant location problems, fixed charge networks, constrained networks, general linear programs with GUB and network substructure, constrained spanning tree problems, traveling salesman problems, and combinations of these. In fact, it has recently been shown that any 0-1 integer programming problems are equivalent to 0-1 generalized network problems.

Network-related problems that arise "naturally" (i.e., those whose formulations make conspicuous reference to a network structure) cover a diverse spectrum of practical applications. These applications include problems in telecommunication, microdata set merging, cash flow,

multi-national currency exchange, manpower planning, waste disposal, water resource management, aircraft refueling, student-professor-classroom scheduling, modular design, ledger and account balance updating, cutting stock, structural and plastic problems of civil engineering, machine loading, aircraft scheduling, off-shore oil well location and drilling, and production and distribution scheduling, to mention only a few.

Setting The Stage

The primary purpose of this paper is to identify recent accomplishments in formulating and solving network and network-related problems, and to propose an integrated sequence of future developments to take advantage of the rich possibilities for new advances and applications in the field. Another purpose is to point out significant questions that have emerged hand in hand with some of the dramatic breakthroughs in modeling and solution techniques during the last few years and to sketch the beginning of some answers.

To achieve these purposes with reasonable conciseness, the next section presents a very brief overview of the recent major developments in network and network-related problems. These developments are then partitioned and discussed in two expository parts followed by a visionary summary. Part I focuses on algorithmic and computational advances for solving linear flow networks. For brevity the discussion stresses the evolution of these advances in the setting of transportation and transshipment problems, and then briefly discusses their related applications for solving other linear network flow problems. Anticipated future developments in the linear network flow area are also discussed in the later sections of Part I.
Part II is devoted to non-linear network flow problems. It attempts to point out the interrelationships between the developments of Part I and corresponding advances in

non-linear network flow problems. The latter sections of Part II briefly discuss some of the major problems thwarting the implementation and application of MS/OR techniques in general.

Recent Happenings

The published research on network and network-related problems prior to 1969 primarily focused on characterizing solution algorithms from a mathematical viewpoint. Very little was done in the way of investigating how broad algorithmic principles should be organized and interrelated to provide efficient computer implementation. Additionally no intensive computer code developments or rigorous empirical studies were presented. Basically the determination of the most effective algorithmic principles, and the development of special techniques for exploiting these principles in a highly efficient manner, were neglected areas.

Similarly on the modeling side, the literature contained almost no attempts to characterize the features or to explore the significance of real world applications. Virtually no work was reported on the examination of alternative model formulations of network-related problems. There were, however, of a handful of elementary model transformation techniques to convert some "near-network" structures into network problems. No one considered whether or not it was possible to devise special formulations that could be accommodated more efficiently (because of the

overall problem size or structural nicities) than the supposedly "inevitable" formulations that first come to mind. For example, in the early literature most network problems were formulated as transportation problems even though the problems could be formulated as transshipment problems with vastly fewer nodes and arcs. This may have been due to some incorrect beliefs that transportation problems could be solved much more rapidly than transshipment.

Things have really changed since 1969! The following developments have all occurred subsequent to this date: (Publication dates cause a number of these to appear as somewhat more recent than the date at which the reported studies were first released):

(1) The first reported transportation and transshipment computer codes based on the dual simplex method (27,32,43),

(2) the first reported techniques for streamlining the out-of-kilter algorithm to enhance its computer implementation (5,64,65),

(3) a number of special updating and labeling techniques and their underlying list structures for improved computer implementation of network algorithms (7,31,34,39,41,47,49,62,71), (reference 49, appearing in 1966, is included among the "more recent" developments because of its pivotal role as a precursor of modern list structure and updating

proposals)

(4) the first reported transshipment code based on the primal simplex method (27),

(5) the first transportation and transshipment codes (2, 50) designed for solving large-scale problems within reasonable solution times.

(6) the first negative cycle transshipment computer code (6),

(7) major advances in primal simplex network computer codes for generalized network problems (35, 57, 61) (in our terminology, the "generalized" network problem is the "flow with gains and losses" problem),

(8) the first non-extreme point generalized network computer code (46),

(9) the first constrained network computer code (38),

(10) the first integer generalized assignment computer code (66),

(11) the first code for solving integer generalized network problems (42),

(12) advanced fixed charge network codes (4, 51),

(13) Frank-Wolfe non-linear network solution codes (48, 59),

(14) advanced location and distribution codes and solution approaches (16, 23, 68, 69).

These code development efforts have been accompanied by rigorous empirical studies to determine the best solution and implementation procedures. Additionally, the past six

years have witnessed the formulation and solution of substantially increased numbers of real world problems as network or network-related problems (9, 22, 23, 25, 37, 44, 45 56, 59). It appears that many researchers are adopting real world implementation as one of the major goals of their efforts, and this is beginning to serve as a driving force in the development of efficient solution and modeling approaches. Indeed, a number of new modeling techniques have appeared which have substantially broadened the class of network-related problems (17, 22, 30, 52, 56), including the discovery (40) that any 0-1 linear program can be modeled as a 0-1 generalized network problem or as a "0-U" transshipment problem (where the flows on certain arcs must equal their upper or lower bounds).

In the area of graphs and "non-flow" networks the "good" (polynomially bounded) algorithm of Edmonds (14, 15) for matching and "degree constrained subgraph" problems have recently been refined for implementation by new labeling techniques (20), and good algorithms have also been developed for matroid intersection problems (58). A substantially more efficient method, called the "quasi-greedy algorithm" (29), has also appeared for solving a subclass of matroid intersection problems that has applications to solving problems in telecommunications, optimal tours (traveling salesman and constrained traveling salesman problems), and vehicle scheduling (21, 28, 29).

Given these advances it is appropriate and necessary to ponder and reflect on a number of questions. For example: What are the implications of these advances for identifying worthwhile directions for future research? What realms of applications have implicitly been opened up that have not yet been fully recognized? What thresholds are about to be crossed, and of these, which are the most important to examine next? What types of interactions are likely to occur between researchers and practitioners due to the unprecedented practical value of many of these advances? What supplementary developments for enhancing the use (and usefulness) of these advances are likely to emerge? The remaining sections of this paper contain some of our more concrete reflections on these questions.

PART I - LINEAR FLOW NETWORKS

Transportation and Transshipment Problems

This section attempts to summarize recent advances in the solution of transportation and transshipment problems. The total work that has been done in this area since 1969 consists of at least 10 man years of work by mathematical programmers and systems analysts. For a more complete discussion of events prior to 1975 the reader should see (9).

The recent developments of simplex computer codes for networks were initiated in two major studies: those of Srinivasan and

Thompson (70) and of Glover, Karney, Klingman, and Napier (32). Both studies introduced a variety of refinements and explored techniques for taking advantage of computational trade-offs that had been left unexamined in the literature to that time. One of the significant findings to emerge from these studies was that the special updating and labeling techniques of the API method (31) (which made use of Ellis Johnson's triple-label list representation (49)) contributed markedly to the efficiency of the procedures of (32), and when subsequently incorporated into the methods of (70) improved their performance by a factor of 2.5. This validated the expectation that a critical factor in the development of any network computer code is the manner in which an effective list structure is integrated with the operations of storing and updating the required information. (Ellis Johnson has said that this realization motivated him to propose the triple label list structure, therefore, he must be credited as the first to foresee the significance of this fact.)

Having decided that the API method was the best updating technique then available, substantial empirical testing was conducted to determine the best starting and pivoting rules to use with simplex based codes (27, 32, 53, 70) and similarly to determine the effect of parameter and problem dimension changes on solution times (53, 70). Following this testing the first primal and dual simplex based transshipment codes were developed by Glover, Karney, and Klingman (27) and similar testing was conducted. These investigations

constituted what could be called the "First Generation" of Modern Transportation and Transshipment Codes (1969-72). This period significantly advanced the speed of solving such problems and saw the first intensive application of the combined skills of mathematical programmers and systems analysts in the network area. These investigations uncovered a number of fallacies in the accepted folklore of that time. For instance, it was shown (27, 32) that the new simplex based transportation and transshipment codes were at least an order of magnitude (i.e., ten times) faster than the best available out-of-kilter and dual simplex codes, and at least two orders of magnitude (i.e., a hundred times) faster than state-of-the-art commercial LP systems. Thus, considerable doubt was cast on the vintage preconception that the out-of-kilter method is inherently superior to a specialized primal simplex method, and on the more recent notion advanced by commercial LP firms that special purpose network codes are not significantly faster than state-of-the-art LP systems.

Another major contribution of this "First Generation" of network advances was the evolution of improved techniques for reducing computer memory requirements of the new codes. For some types of applications, this contribution was certainly as important, if not more important than the advances in problem solving speed. By means of the effective integration of list structures and updating techniques, memory requirements were more than halved during this generation. Thus, the

term "half life generation" was conceived. This term also conservatively characterizes the increase in solution speed from 1969-72, independent of the dramatic initial advances in 1969 and early 1970.

All of the codes developed during this period were programmed in FORTRAN and were in-core computer codes (that is, codes which keep all problem data in central memory) except for the in-core out-of-core transshipment code by Karney and Klingman (50). This code incorporated the most effective techniques of this period into a FORTRAN code capable of solving problems from the domain of large-scale applications. The development of this code was motivated by a manpower planning problem of the U.S. Navy that contained 450,000 arcs, which was vastly larger than any large-scale OR application of its type previously considered. The code succeeded in solving this problem in 26 minutes of CP time on an IBM 360/65.

This First Generation of network advances also included the first major streamlining of the out-of-kilter algorithm for computer implementation by Barr, Glover, and Klingman (5). New techniques were devised to handle the labeling and updating portions of the algorithm. These produced a six fold improvement in solution times over previous out-of-kilter codes. Hogan (48) later implemented these techniques for the U.S. Air Force and obtained comparable results. These new procedures for implementing the out-of-kilter approach were, however, still unable to overtake the efficiency of the best

simplex-based approach either in solution speed or in memory requirements.

During this same period, Bennington (6) also extended Klein's negative cycle method to transshipment problems, and developed a computer code based on this extension. Although a definite advance in the implementation of methods of this kind, the "negative cycle" code was found to be at least an order of magnitude slower than the best simplex-based approach, and to require at least twice as much computer memory (54).

Following in the aftermath of all these developments, and taking advantage of them, the Second Generation may be conceived as originating in 1973 with the development of a new type of list structure and updating scheme for storing and manipulating the spanning tree basis. This new approach by Glover, Klingman and Stutz (39), called the Augmented Threaded Index (ATI) method, provided both increased efficiency and reduced memory requirement over the previously best API method. Mulvey (63) elaborated on this development by integrating the ATI approach with the distance function concept of Srinivasan and Thompson (71) (which had been introduced in the First Generation but did not play a central role in code design until the Second). Mulvey's integration of the distance function and the ATI approach improved solution times another 10 % at the expense of an additional node-length array of computer storage.

Another significant contribution of the ATI method stems from the fact that it became possible to design an in-core out-of-core simplex transportation code which only uses four node length arrays to maintain and update all basis information (if the distance function is not used). This advance is of paramount importance for solving the extremely large problems that sometimes arise in practical applications. One of these is a micro-data set merger problem recently formulated by the U.S. Treasury. This problem is a transportation problem which, after calling out extraneous links, contains 50,000 nodes and 62 million arcs. The U.S. Treasury requires problems from this class to be solved several times a year on a UNIVAC 1108, and contracted Analysis, Research, and Computation (ARC), Inc. to design and implement a code capable of meeting these requirements. By using the ATI labeling and updating method, such a code (2) has been developed in FORTRAN and to date has run problems with 5000 nodes and 625,000 arcs in less than 4 minutes of CP time (and 9 minutes total processing time) on a UNIVAC1108. The full problem size capability of this code will be tested in the near future when the U.S. Treasury completes the process of generating the full size problems.

The most significant finding of this "Second Generation" in the area of pivot selection strategies was made by Mulvey (63), who showed that solution times can be substantially reduced for problems with more than a thousand nodes by using a special form of a "candidate list" pivot strategy.

Mulvey's candidate list is constructed in accordance with the pivot rule found to be best for smaller problems - that is, the small problem pivot rule is used to select several nonbasic arcs to be designated as candidates and then the actual choice of the "incoming arc" is restricted to these candidate arcs. After a predetermined (or dynamically determined) number of pivots - or whenever the candidate list contains no more members that are eligible to be the incoming arc - a new candidate list is constructed. The best candidate list strategies have reduced solution times by as much as 60 %.

During the early part of this Second Generation, McBride and Graves (47, 62) developed a general technique called the factorization method. The first implementation of this method in the network setting (reported in (47, 62)) demonstrated the possibility of specialized application in this area. The implementation, however, proved to be substantially slower than the API and ATI procedures and required even more computer memory than the initial implementation of the API approach. These limitations were of course due in large part to the experimental nature and lack of refinement of the first implementation of any approach. Indeed, a later improved implementation by Graves (not reported in the literature) showed that the method was susceptible to substantial improvement, although it remained twice as slow as the implementation of the ATI procedure reported in (27) and continued to require more computer memory,

when both procedures were tested on the same computer and same problems at General Motors, Inc. Nevertheless, the factorization ideas contain a good deal of ingenuity, and future implementations may be found that are superior to those developed so far.

The Second Generation was also attended by new efforts to streamline the out-of-kilter method. Several researchers, including Jensen and Bhaumik (46) and Rao and Fong (65), proposed using the API techniques to retain usable portions of the labelling tree. This idea improved solution time by a factor of 3-5 over classical out-of-kilter codes at a cost of only two additional node-length arrays. Its use in conjunction with the earlier streamlining of (5) has not been tested. We anticipate this will be done in the near future. Since the earlier streamlining produced a six fold increase in speed, it seems quite possible that the combination of these techniques may result in out-of-kilter codes that are competitive with or even faster than simplex-based codes. Recent investigations that may provide insight into these matters are being conducted by Aashtiana and Magnanti (1). If the out-of-kilter codes can be made as efficient as those based on the primal simplex method, it is probable that their primary application will be on small or moderate sized problems. This is due to the fact that the out-of-kilter approach seems to require a substantial amount of computer memory and does not appear to lend itself to an efficient in-core out-of-core implementation. However, it is conceivable that a

hybrid combination of the out-of-kilter and primal simplex approaches will be developed that overcomes some of these limitations.

This concentrated burst of activity that we have called the Second Generation was quite short, and in referring to it we have indeed included mention of developments that properly carry over to the present "Third Generation". We choose to demark the beginning of the Third Generation as coinciding with the development of a third updating and labeling scheme for improving the implementation of network algorithms. On the basis of sound analytical arguments, it is quite likely that all updating and labeling schemes for simplex network codes will henceforth only augment the information kept by this approach. This new approach by Glover and Klingman (41) which appeared in mid-1974, augments the ATI method with two new functions, called the "cardinality" and "last node" functions. The use of these functions in conjunction with the ATI method makes it possible to update flows and node potentials at each iteration with a marked increased in efficiency. Further, both of these functions together require less work to update than the distance function. As a final bonus, the "cardinality function" can accommodate all of the relevant functions filled by the distance function, and hence can replace it. The net result of all these advantages produces a substantially improved procedure for implementing basis exchange operations.

These latest developments are currently being implemented and tested by ARC, Inc. in conjunction with Barr, Glover, Karney, Klingman, Mulvey, and Stutz. Bradley, Brown, and Graves have recently tested the cardinality function as an alternative to the distance function and have obtained some excellent improvements in solution times (7).

The work of Aashtiani and Magnanti (1) mentioned earlier actually belongs to the Third Generation. These developments bear watching. In addition, Shapiro (67) has noted a previously overlooked distinction between the out-of-kilter method and the primal-dual method that may hold promise for exploiting the latter method by subgradient optimization and by the techniques of (41). Thus the Third Generation promises to produce several enhancements in the implementation of the primal simplex and out-of-kilter approaches and may initiate a new generation of hybrid solution approaches.

Future Developments for Transportation and Transshipment Problems

In spite of the notable gains in network solution techniques since 1969, there is still a major area that remains to be explored. This is the area of designing appropriate techniques for accommodating and taking advantage of degeneracy. Computational testing has shown that approximately 90 % of the pivots for transshipment problems with more than 1000 nodes are degenerate. Thus far, no computational schemes have been developed to respond to this situation, either

by an effort to circumvent degenerate pivots or to make them judiciously.

However, this area is being seriously researched, and promising results should be obtained within the near future. It is estimated that an effective scheme for handling degeneracy could easily reduce solution times by another 50 %.

Another development which should prove to be a significant future advance in mathematical programming is the implementation of network algorithms on mini-computers. Given the dramatic recent improvements in the speed and memory requirements of network methods, and the major advances in mini-computer design, it now appears possible to implement relatively efficient in-core out-of-core transportation and transshipment codes in an assembly level language on most mini-computers. This belief is partly based on the work reported in (2, 50) which indicates that simplex based codes do not suffer undue increases in solution times by keeping the problem data in external computer memory.

The emergence of network codes on mini-computers would afford the potential to substantially increase the use of mathematical programming in the real world. In particular, it may well be possible to make such codes available on mini-computers that cost about $ 15,000 and that include

cassette storage and a cathode ray tube (CRT) display. Such a development would make it possible to demonstrate OR techniques easily and conveniently to managers in their own offices, at management seminars, and in other countries. Further, since the computer could be dedicated to network applications, it would be feasible (and highly desirable) to design the cperating system to minimize the human engineering aspects of problem-solving -- that is, to minimize the difficulties of entering, modifying, and verifying problem data, passing the problem data to the solution code, interpreting the output, and so forth. With a dedicated system such problems could be almost nonexistent to the user.

Lack of satisfactory treatment of the human engineering aspects of problem-solving constitutes one of the major obstacles to a more widespread use of mathematical programming today. The next decade will produce major advances in this area. It is envisioned that such advances will particularly occur by interfacing optimization routines with management information systems for sophisticated users. The counterpart of these advances for small companies is the development of minicomputer capabilities. The rationale for this conjecture is that the major data processing hassles for small companies that cannot afford a large computer are the difficiulties of using commercial time-sharing systems. The inconvenience and cost of buying time and associated

terminal equipment, as well as overcoming the technical aspects of using such a service, discourage the small companies from using OR to accomplish tasks currently done in some workable fashion with pencil, paper, and desk calculators. Thus, while OR is at present used primarily by large companies and government agencies for "big" decisions, the envisioned coupling of inexpensive mini-computers and network algorithms suggests that the applications of sophisticated OR techniques for "small" decisions will soon be possible.

Transfer of Transportation and Transshipment Technology to Related Problems

Assignment Problems

With the exception of the work by Rao and Fong (65), very little research has been directed toward specializing the recent network innovations to the solution of assignment problems. Consequently the most efficient way to solve assignment problems is currently unknown. Within the next few years this question will be resolved. It is also suspected that current simplex problem-solving technology can be modified to respond in a uniquely appropriate fashion to the "massive degeneracy" that is characteristic of assignment problems, as well as to reduce computer memory requirements for this class of problems. Further it is predicted that such specialized simplex based procedures will prove to be the fastest solution approach.

Generalized Networks

Currently the updating and labeling techniques for transportation and transshipment problems are being extended to generalized network problems also called "weighted networks" or "networks with gains". The fundamental relationships by which such extensions can be carried out effectively are developed in Glover, Klingm n, and Stutz (34), which shows how the diverse basis updating configurations (of which there are more than a dozen separate cases) can be consolidated into a single "general case". This result builds upon the early work of Balas and Hammer (3) who characterized the basis structure for this class of problems as a forest of "1-trees" (i.e., trees augmented by an additional arc). Additional computational simplications of the algorithmic steps for generalized networks have appeared. For example, Maurras (61) and Glover and Klingman (36) indicate how to simplify basis updating calculations by characterizing an appropriate sequence for tracing out paths in the one-trees and for proceeding through all nodes in a subtree of a one-tree.

These events led to the development of efficient simplex-based codes for generalized networks by Glover, Klingman, and Stutz (35), Langley (57), and Maurras (61). Present computational results indicate that generalized network problems require about four times longer to solve than transportation and transshipment problems of similar dimension. However, a series of algorithmic and implementation advances in the next several years should substantially

reduce the already impressive solution times of such problems. For example, generalized networks of 1000 nodes and 7000 arcs are currently being solved in less than 60 seconds on a CDC 6600 by the method of (35). It is believed this time will be easily cut in half by adapting the new labeling and updating techniques for pure networks to generalized networks by means of the relationships developed in (34) and by exploiting degeneracy. Degenerate pivots occur as frequently in generalized networks as in pure networks.

A new non-simplex solution approach for a class of "maximum flow" generalized networks has been developed by Jensen and Bhaumik (46). The approach is based on an extension of the Busacker and Gowen network flow algorithm (8). A computer code based on this approach appears to be quite efficient, although slower than the simplex-based code of (35). The principal limitation to be overcome for this type of approach is the requirement of substantially more computer memory than the simplex-based approach.

Constrained Network

By a constrained network we mean an assignment, transportation, or transshipment problem which also includes extra linear constraints. Klingman and Russell (52) have developed a special basis compactification procedure for this class of problems that maintains all of the network as the "implicit" portion of the basis. The operations involving the implicit portion of the basis are simply carried out

as they would ordinarily be for the underlying network. Thus it is possible to take advantage of the efficient procedures previously described for solving network problems. Preliminary computational implementations by Glover, Karney, Klingman, and Russell (30, 38) indicate that this procedure is highly efficient for solving problems with one or two extra constraints. Such problems typically required only twice as long to solve as the underlying network problem. A prototype code using related ideas to handle more than two extra constraints has recently been developed by Chen and Saigal (10) although without attempting to incorporate the essential refinements required to take full advantage of the network structure or to optimize the computer memory requirements. We strongly anticipate that refined codes to handle larger numbers of extra constraints will be developed in the near future. The results of (38) suggest that these codes will be much more efficient than general linear programming codes for solving this class of problems.

The importance of the basis compactification area for solving constrained network problems is difficult to overrate. It will probably be the focus of numerous active investigations and will produce the next breakthrough, similar to the GUB breakthrough, in the developments of linear programming codes. This belief rests on the following observations:

a) GUB is a specialization of the network basis compactification procedures of (52). This can be seen by observing that GUB constraints form a network by simply scaling and summing the GUB constraints to form another constraint. When this additional constraint is added to the GUB constraints, the system of constraints is a network.

b) The network basis compactification extends GUB in several important ways. First it allows the GUB variables to have individual upper bounds. Second it eliminates the nonoverlapping variable requirement of GUB.

c) The number of real world LP problems which have network substructures is quite large. Further the IP problem manipulation schemes for obtaining stronger LP relaxations (73) often induce network structures. For example, consider the constraint $x_1 + x_2 + x_3 - 3x_4 \leq 0$ where x_1, x_2, x_3, x_4 are 0-1 variables. It is well known that replacing this constraint by the constraints:

$$s_1 + x_1 = x_4$$
$$s_2 + x_2 = x_4$$
$$s_3 + x_3 = x_4$$

yields a stronger LP relaxation. Note however that by performing elementary row operations this latter set of constraints is equivalent to:

$$0 = -x_1 + s_1$$
$$0 = +x_1 - x_2 - s_1 + s_2$$
$$0 = x_2 - x_3 - s_2 + s_3$$

Subtracting these constraints produces $0 = x_3 - s_3$. Appending this equation to the others yields a network. Thus by combining the IP problem manipulation ideas with the network basis compactification feature the working basis of the LP problem solved is reduced by 1. Thus the addition of such features to commercial LP systems would substantially overcome the problem expansion concern of many practitioners when using stronger IP/LP formulations.

We also anticipate that several variations of the basis compactification scheme will be developed to handle constrained network problems with special classes of extra constraints and constrained network problems which contain a few ordinary L.P. variables. Extensions to handle generalized networks with extra linear constraints are in fact relatively straight forward, and implementations will no doubt soon appear. Compact basis techniques that are particularly efficient for plant location and sequencing problems that arise in IP settings have also recently been developed in (24).

Modeling

As a result of the projected algorithmic and implementation activities for generalized and constrained network problems, we foresee a flurry of modeling advances for these problems. In our opinion, such problems have a vastly richer domain of application than ordinary networks: e.g., examples include cash flow problems, multi-commodity problems, waste disposal problems, water resource management problems, plastic

problems of civil engineering, production scheduling problems, machine loading problems, goal programming networks, stochastic and constrained regression problems, and multi-national currency exchange problems, to mention only a few.

An important educational function that should soon be adopted by members of the Operations Research and Management Science fields is to train and guide modelers and decision makers in visualizing their problems within network-related frameworks -- particularly those of generalized network or constrained network problems when this is possible. A trend is already underway (40, 42, 44) toward identifiying problems that can be framed cost-effectively as network-related problems. Network-related problems provide natural formulations for a vastly larger number of real world problems than previously suspected. At the same time, researchers are discovering that a variety of other real world problems can be easily transformed into this format (42, 44). Such models and formulation possiblitities have been virtually unknown in the literature before now because of the lack of efficient ways to solve such problems as a class distinct from general LP models.

It is further believed that the effort to help people think of their problems in a network-related format will have far reaching and beneficial consequences in two ways.

First, this type of visualization will enable decision-makers to formulate their problems initially by means of drawings and diagrams. This is much easier for the non-technical person than trying to formulate his problems in terms of equations. The result will be to provide increased recognition of the pervasiness and sign. ficance of these types of problems, and thus to enlarge the number of important real world problems that are being solved by OR methodology. The second and probably more significant consequence will derive from the fact that these types of problems can often be solved with dramatic success, even in large-scale applications. Consequently, a much greater appreciation of OR by industry and government will come about, permitting the widespread disillusionment that followed on the heels of many attempted LP and IP implementations to be reversed. Indeed, it seems conceivable that such developments may spearhead a new liason between OR and its users. We have personally had first hand experience of this upsurge of interst by industry and government when such modeling and solution approaches are used. Recent real world applications which have been successfully solved via these modeling and solution approaches are summarized in (44).

PART II - NONLINEAR NETWORK FLOW PROBLEMS

The modeling and solution on nonlinear network flow problems is almost wholly an uncharted realm of the network area. For expository purposes the discussion on nonlinear network flow problems is divided into four separate but interrelated parts: modeling advances, methodology and implementation horizons, human engineering advances, and standards for code developments.

Modeling Advances

The breadth of the problems encompassed by this area has only recently come to be realized. To demonstrate the potential of the modeling area, and to help the reader better understand some of the statements and predictions to be made later in Part II, a simple variant of a modeling technique given in (40) for transforming a 0-1 IP problem into a 0-1 generalized network will be illustrated. Consider the problem:

$$\text{Minimize } 3x_1 + 4x_2 + 7x_3$$

subject to

$$\begin{aligned} 2x_1 + 0x_2 + 4x_3 &= 4 \\ -1x_1 + 10x_2 + 0x_3 &= 10 \\ 4x_1 + 0x_2 + 0x_3 &= 4 \\ 0x_1 + 6x_2 - 1x_3 &= 5 \\ x_1 , x_2 , x_3 &\in \{0,1\} \end{aligned}$$

To construct a 0-1 generalized network formulation of this problem, create nodes (A), (B), and (C) for the variables x_1, x_2, and x_3, respectively. Also create a node (O) and connect node (O) via arcs to nodes (A), (B), and (C). The flows on these arcs will correspond precisely to the values assigned to the corresponding variables (see Fig. 1). Thus the costs attached to these arcs are the same as those in the objective function for the problem in its original form.

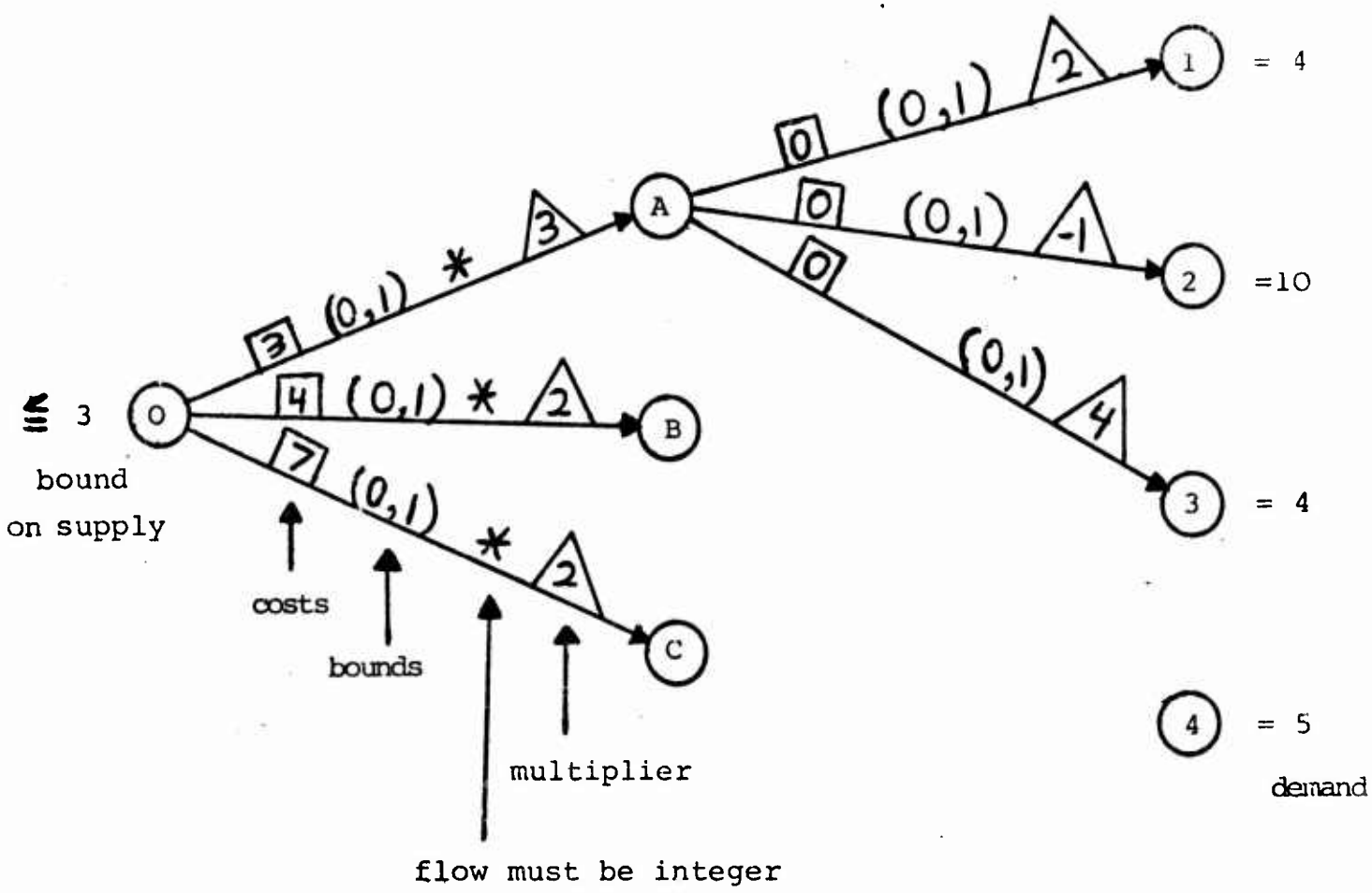


Figure 1

Similarly, each of these three arcs is given an upper bound of one and a lower bound of zero, and a multiplier (i.e., the factor by which flow is transformed as it traverses the arc) equal to the number of constraints of the original problem in which the corresponding variable appears with a nonzero entry. Thus, for example, the arc (O,C) receives a multiplier of 2, because x_3 appears in two constraints.

The next step is to create a node for each of the problem constraints, which we will designate by (1), (2), (3), and (4). The earlier "variable nodes" (A, B and C) are then connected to these "constraint nodes" in the following manner. Node A is connected via an arc to each of the nodes (1), (2), (3), and (4) for which x_1 has a nonzero coefficient in the associated constraint. These "variables to constraint" arcs are each given a cost of zero, lower and upper bounds of O and 1, and a multiplier equal to the coefficient of x_1 in the associated constraint (as shown in Figure 1). The same types of connections are made for x_2 and x_3 by reference to their associated nodes (B) and (C) (although these connections are not illustrated). Finally, node (O) is given a supply bounded by the total number of O-1 variables, and each of the constraint nodes is given a "demand" equal to the right hand side for this constraint in the original problem. By restricting each arc leading out of nodes (O) to have a flow of O or 1 (indicated by the asterisk in Figure 1) it is easily seen that this problem is equivalent to the original O-1 problem.

It is important to note that the network formulation has actually the same number of integer variables as the original formulation. The number of continuous variables added to the problem depends on the particular transformation used. (The simple approach illustrated in Figure 1 introduces more arcs and continuous variables than necessary.) In fact, just as there are different IP formulations for the same problem, we are finding that there are different types of network-related formulations (not all of which are "obviously" related to the type of formulation just illustrated). For example, by creating additional nodes and arcs, it is also possible to transform 0-1 IP problems into "0-U" transshipment problems where the flows on certain arcs must be equal to their upper or lower bounds (40). Also in the case of problems with special structure, such as set covering and set partitioning problems, there are several specializations and simplifications of these basic ideas that hold promise of permitting more effective treatment by appropriately designed algorithms. We predict that alternative formulations, manipulations and relaxation schemes based on these ideas will prove to be a fruitful area of investigation during the next few years.

It is clear that the preceding transformation techniques extend quite naturally to accomodate mixed integer 0-1 problems where the continuous part of the problem is a transportation, transshipment, or generalized network problem itself. Many important real world applications have such a "mixed" structure, among them are a variety of energy models, plant location models, and distribution models. An

energy model which has this network-related formulation is the subject of one of our current problem-solving investigations

The trend indicated earlier toward identifying problems that can be cost-effectively framed as network-related problems will undoubtedly be especially pronounced for those problems attended by auxiliary integer (or "discrete") restrictions. Once people are educated to visualize problems in network-related frameworks, the initial "recognition" of many IP and LP problems will be made far easier. Indeed, rather than contemplating ways to transform systems of constraints into a network-related structure we may be challenged to transform "network representations" of problems into alternative mathematical representations -- or to devise new ways of tackling these network representations directly. It is convenient to distinguish these types of network representations from those that correspond to "actual" network problems by referring to them as NETFORMS. (For example, the diagram of Figure 1 may be regarded as one type of NETFORM.)

This change of orientation by which OR problems are directly visualized and represented as NETFORMS should have highly beneficial effects along several dimensions. First, OR people will be able to use these NETFORMS to explain their models to the users more effectively. Second, the non-scientific user can be more easily taught how to formulate his problems via such diagrams and pictorial representations which OR people can then analyze and refine. These effects have the advantage of inducing the management scientist and the user to interact more fully, thereby leading to clearer mutual understanding of

the problem and of the potential meaning and usefulness of its solution. In fact, we suspect that books will appear in the future with a variety of applications discussed in the NETFORM framework. Still more importantly, it is anticipated that a standard set of formulation symbols and flowchart diagrams will appear to make the interpretation of alternative NETFORMS more precise. (For example, the symbolic conventions of Figure 1 serve unambiguously to define the problem associated with the NETFORM.) We envision this standardization to be similar in principle to the advent of Structured Programming in Computer Science. Such a standardization may conceivably lead to new insights about methods for solving particular classes of problems, as well as to more convenient ways to formulate them.

Methodology and Implementation Horizons

It may seem that the presumed value of transforming "non-network" problems into network-related problems is highly speculative except possibly for the visualization advantages that may be provided by associated NETFORMS. However, we have applied a transformation technique such as illustrated in Figure 1 to a 0-1 formulation of a U.S. Air Force pilot training problem with remarkable success. The 0-1 formulation is a problem with 460 zero-one variables and 730 constraints. The network-related formulation has 460 zero-one variables, 2200 continuous variables, and 900 nodes. This would represent a fair increase in size as an LP problem, but provides a relatively small network problem. The NETFORM was solved

using a specialized branch and bound procedure with transshipment subproblems. The optimal solution was found and verified after solving only 19 subproblems, requiring a total solution time of 10 seconds on a CDC 6600. This is exceptionally fast for a 460 x 730 zero-one problem. Consequently, we feel that there exists an excellent potential for finding efficient solution procedures to many IP problems by making use of such transformation techniques. However, procedures that are appropriately tailored to take advantage of network-related structures that arise from such transformations definitely do not exist today and their derivation will require extensive research.

The most successful approaches for solving many of these problems will undoubtedly make extensive use of the existing linear network codes and the new ones foreshadowed in Part I, due to the remarkable efficiency of these codes for handling network structures. Thus, the discussion will focus on such approaches. It is certainly true that other kinds of approaches can be more effective for solving some network-related problems. A notable illustration is the method of Ross and Soland (66), which uses a Lagrangean relaxation approach involving knapsack subproblems, to solve a "generalized assignment" problem. This method has proved capable of solving problems with 20 origins, 200 destinations, and 4000 0-1 variables with a median solution time of 1.5 seconds, and has also solved 20 x 1000 problems with 20,000 0-1 variables in 30 seconds. Thus, we expect to see similar specialized "non-network" solution approaches prove to be highly effective for solving some

network-related problems.

The development of efficient codes to solve constrained network problems combined with the existence of different types of network-relaxations within formulations for network-related problems, provides an enormous set of alternative solution approaches. An interesting subset of these approaches consists of the hybrid branch and bound/ cutting plane approaches. For this subset, research and computational testing is needed to determine which of the relaxations, problem manipulations, branch and bound methods and cutting plane approaches are best in combination. In particular, we feel that research is needed to determine trade-offs between the following areas:

(1) Strength of relaxation. The importance of testing this is clearly pointed out in the paper by Williams (73). Better bounds imply less branching and will therefore shift the balance of work to the efficient linear network flow code.

(2) Branch and bound choice rules and special penalty calculations tailored to networks. These rules and calculations should not just copy those applied in LP settings, but should be based on considerations that are unique to network structures. The best intersection of available penalties, choice rules, and formulations must be ascertained.

(3) Heuristics tailored to network formulations. Heuristics often provide good starting solutions for the branch and

bound procedures.

(4) Cut specializations. That is, calculating cuts from the network structure and introducing them back either in the form of an augmented network structure or as side constraint. Research is needed to answer such questions as: How much can be incorporated into the network and how much remains to be treated as side constraints? Can the side constraints be induced to have a special structure? Can a block of cuts be added that introduces more network structure but keeps the side constraints at the same level as for just one cut? How can the network be expanded and contracted during these operations in an efficient manner? (Art Geoffrion has said that he believes the general area of determining how much of a particular problem can be represented by a "true network" and how much must be left over constitutes an important domain of research, independent of the special considerations discussed here.)

(5) Computer decomposition and problem decomposition. Computer decomposition (or algorithmic decomposition) refers to designing the computer program to handle some of the logical constraints explicitly in the choice rules and specialized calculations rather than treating all constraints directly in the problem. By this means it is often possible to reduce the size and complexity of the subproblems. This is an area that is only beginning to be probed, and whose significance should be much better appreciated in the near future.

In order to implement and evaluate alternative algorithms properly, special research must be devoted to developing data organization and manipulation schemes that are most effective in specific kinds of applications. This is particularly important when linear network codes are used as subroutines in a master program, as for example in solving various classes of "integer constrained" generalized networks. In such circumstances, the organization of the data interfaces between the program, which also influences the internal workings of each, can have a profound effect on overall efficiency.

This is an area which has not been fully considered in the design of network solution codes. We were rather abruptly confronted with this fact in our work to design a fixed charge transportation code (4) and a mixed integer generalized network code (35). Data organization schemes that we had previously regarded as both convenient and natural soon disclosed a variety of weaknesses in these settings. The development of procedures to coordinate data organization and manipulation procedures to produce the most effective methods for particular problem classes will require innovations both of an "algorithmic" nature and of a "programming" nature. Nevertheless, the payoff from such innovations should be high. As discussed in Part I, the integration of such procedures with the basic processing steps is absolutely essential to accurately evaluate the solution efficiency of alternative methods. For example, it can be said that the out-of-kilter method was thought to be better than primal simplex-based

network approaches because of a failure to conceive of an appropriate integration scheme for the primal approaches. The problem of coordinating data organization and manipulation schemes is investigated in (5) in the context of fixed charge transportation problems. Careful attention to this issue in the fixed charge setting produced a solution code that is able to solve problems with 200 nodes, 600 fixed charge variables, and 3000 continuous variables in 10-20 seconds on a CDC 6600. Thus we can not overemphasize the importance of coordinating organization and manipulation schemes between levels of complex solution routines.

In response to this issue, we ultimately envision the development of linear network systems with their own control and command languages. By means of these languages, network systems can be integrated into master routines as subroutines in such a way that the master program need not be at all concerned about "how" and "where" to modify the network data. The master program will simply pass commands like MODIFY FUNCTIONAL BY 2 TIMES GRADIENT or CHANGE DUAL W2 TO 3 (as in subgradient optimization or Lagrangean relaxation routines) and the network system will do this in the most effective manner. The design of such command systems will require considerable thought to anticipate major operations that users will want to perform, and it may well be that different command systems will be required in different situations. We should point out that these speculations are not altogether fanciful. The interactive and batch UKILT system designed by Barr, Glover, Klingman, and Stutz for UNIVAC, Inc. (72), and the CAPE system designed

by Glover, Karney, and Klingman for the U.S. Navy (38) are rudimentary forerunners of such systems.

In this same connection, we anticipate that such systems will indeed be designed like UKILT to accommodate both interactive and batch processing. The conveniences of interactive processing can substantially increase user productivity and effectiveness in all phases -- modeling, problem modifications, analysis of solution statistics, data management, hand tuning of algorithmic parameters (e.g., branching rules and penalty calculations), etc. These features of interactive processing should be particularly valuable for testing alternative network-related formulations or relaxations in conjunction with specific solution strategies.

Human Engineering Advances

The human engineering aspects of a mathematical programming system involve the phases of problem solving in which the user "interacts" with the system. Thus, for example, human engineering includes the processes by which information and problem data are acquired by the user to build or implement a model, and the subsequent processes by which information and data are returned and implemented by the user in the form of a solution. Effective handling of the human engineering phases of problem-solving is vital to the full and successful application of OR. This is evident even in the restriced activities of human engineering that have to do with accessing and reporting information to a user's data base. Inefficient processing to accommodate such activities can greatly increase

the total computer time required to solve a problem and more importantly can greatly increase the human time and effort required to use OR techniques. And thus substantially discourage its use. We believe this area currently presents a major stumbling block to many linear network applications. For example, as mentioned earlier, Analysis, Research, and Computation (ARC), Inc. designed a code to solve transportation problems with 50,000 nodes and 62 million arcs for the U.S. Treasury (2). However, the Treasury is now finding it very difficult to generate such problems in a reasonable amount of computer time. Test problems that the ARC code solved in 4 minutes of CP time required several hours of computer time to generate! Consequently, we believe highly sophisticated data-based management information systems will come to serve as natural accompaniments to network-related solution codes. For major applications, combined packages of this sort are overdue. Highly specialized graphic display packages linked to network-related LP and IP solution codes as input and output options are also foreseen. The appearance of such display packages could be hastened with refinements and standardizations in the presentation of NETFORMS as a convenient way to capture the "visual essence" of diverse problem structures.

Standards for Code Development

The gradual escalation in the implementation and testing of computer codes for solving real world mathematical programming problems and the anticipated development of a host of non-linear

codes is placing strong demands on the MS/OR community to establish standards and guidelines. However, before the MS/OR community can do this, it must fully accept a concept which so far seems alien to it: that the application of scientific principles to develop, test, and evaluate computer implementations is an appropriate and indeed valuable form of research. The acceptance of this concept is essential in order to guarantee that highly qualified researchers can be recruited and supported to make significant contributions to code development. We feel that it is important to establish forums at conferences and circulate detailed newsletters to discuss the creation of standards in an open and interactive setting. Newsletters are also needed in order to communicate the latest findings that have been validated by the application of such standards. Current computer codes are able to perform fantastic feats in seconds but our experience over the years is that even knowledgeable and concerned researchers are often two years behind in becoming aware of these events. This "lag time" is definitely unacceptable for computer research. Further, after an implementation paper goes through the current refereeing process, the authors are often required to drastically condense details of the implementation phase. Often several revisions are required which force the author to present sketchy descriptions of results and which cause the results of the paper to become obsolete before publication. Those who have met with these kinds of experiences will undoubtedly agree that the area of computer implementations needs its own set of guidelines, particularly to determine the significance and suitability of various types of information to be reported in the literature.

More importantly, guidelines need to be established for testing, validating, and certifying OR computer codes. This is an extremely difficult task which has not been fully resolved in other related computer fields such as numerical analysis. However, we need to begin to seriously devote attention to creating guidelines for OR codes.

Before firm testing standards can be applied to researchers, however, data bases of real world problems must be established. In addition, conveniently standardized data formats are required to allow researchers to have "true access", rather than merely hypothetical access to such data bases as sources of test problems. The standardization of magnetic tape formats used to distribute problems should also be required. Thus, for example, the data format should allow users to conveniently test different network-related formulations of IP problems, and the tape format should insure readability on the large machines of major computer manufacturers.

The procedure for certifying that a code "executes correctly" needs to be carefully and thoroughly examined. Key parameters to identify the range and characteristics of problems for which the code executes correctly need to be established. Also, it is important to provide clear definitions of code efficiency and how it is to be evaluated for both general and specific problem classes. Currently, certification and validation of mathematical programming codes are non-existent. Each researcher's own code is the "most" efficient and accurate for solving his test problems!

We fully expect to see all phases of the experimental development, testing, and evaluation of computer codes accepted as research by the MS/OR profession within the next couple of years. And in spite of the acknowledged difficulties, we look forward optimistically to the imminent appearance of formal sessions, forums and newsletters to discuss such topics as: setting guideline policies, efficiency questions, major limitations of current data structures, maintenance and distribution of test problems, major obstacles to particular implementation approaches, etc. We feel these developments are essential if OR is to remain a viable decision making tool in the computer age.

Horizons

In summary the future of network and network-related research could integrate the modeling and solution phases of the MS/OR discipline utilizing the most recent advances of the computer age. We envision this integration as a multifaceted interaction between modelers, algorithmic researchers, systems analysts, computers, and computer graphics. This interaction will need, however, to take place in a cohesive and productive environment which blends the best of both man and machine in order to reap maximum application benefits. It is our belief that this integration is not merely beneficial but essential if OR/MS is to become a major vehicle for solving pressing industrial and governmental problems. The integration is envisioned along the following lines. The modeling and user communication phases of MS/OR could be significantly advanced by using a

NETFORM visualization of the problem. The NETFORM concept of pictorially visualizing a problem would help the non-mathematica manager and administrator to appreciate and use MS/OR techniques for problem-solving and decision-making. Additionally a natural integrate of NETFORM with the computer age exists via graphic display facilities. Finally, it is our belief that a viable solution approach for many large scale integer programming applications can be found by an integrate of the NETFORM concept with the recent advances in network solution procedures. The discovery of these solution procedures will, however, only be accomplished by co-ordinating computational testing and theoretical research with an integration of talents of system analysts and mathematical programming specialists.

REFERENCES

1. H. Aashtiani and T. L. Magnanti, "Solving Large Scale Network Optimization Problems", Presented at ORSA/TIMS conference, Chicago, April 1975.

2. Analysis, Research, and Computation, Inc., "Development and Computational Testing on Large Scale Primal Simplex Network Codes", ARC Technical Research Report, P. O. Box 4067, Austin, TX 78765 (1974).

3. E. Balas and P. L. Ivanescu (Hammer), "On the Generalized Transportation Problem", Man. Sci. 11, 188-202 (1964).

4. R. S. Barr, F. Glover, and D. Klingman, "Implementation of an Efficient Fixed Charge Transportation Code", Presented at Joint ORSA/TIMS conference, San Juan, Puerto Rico (1974).

5. R. S. Barr, F. Glover, and D. Klingman, "An Improved Version of the Out-of-Kilter Method and a Comparative Study of Computer Codes", Mathematical Programming, 7, 1, 60-87 (1974).

6. G. E. Bennington, "An Efficient Minimal Cost Flow Algorithm", Man. Sci., 19, 9, 1021-1051 (1973).

7. G. Bradley, G. Brown, and G. Graves, "A Comparison of Storage Structure for Primal Network Codes", Presented at ORSA/TIMS conference, Chicago, April 1975.

8. R. G. Busacker and P. J. Gowen, "A Procedure for Determining a Family of Minimal-Cost Network Flow Patterns", ORO Technical Report 15, Operations Research Office, Johns Hopkins University (1961).

9. A. Charnes, F. Glover, D. Karney, D. Klingman, and J. Stutz, "Past, Present, and Future of Development, Computational Efficiency, and Practical Use of Large Scale Transportation and Transshipment Computer Codes", Research Report CCS 131 Center for Cybernetic Studies, University of Texas, BEB 512, Austin (1973), to appear in Computers and Operations Research, 2, 2 (1975).

10. Chen and Saigal, "A Primal Algorithm for Solving a Capacitated Network Flow Problem with Additional Linear Constraints", Presented at ORSA/TIMS conference, Chicago, April 1975.

11. J. B. Dennis, "A High-Speed Computer Technique for the Transportation Problem", JACM, 8, 132-153 (1958).

12. E. A. Dinic, "Algorithm for Solution of a Problem of Maximum Flow in a Network with Power Estimation", Soviet Math. Doklady, 11, 5 (1970).

13. J. Edmonds and R. M. Karp, "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems", JACM, 19, 248-264 (1972).

14. J. Edmonds, "Paths, Trees, and Flowers", Canadian Journal of Mathematics, 17, 449-467 (1965).

15. J. Edmonds, "Maximum Matching and the Polyhedron with 0-1 Vertices", Journal of Research of the National Bureau of Standards, 69B, 125-130 (1965).

16. L. B. Ellwein and P. Gray, "Solving Fixed Charge Location-Allocation Problems with Capacity and Configuration Constraints", AIIE Transactions, III, 4, 290-298 (1971).

17. J. Evans. J. Jarvis, and R. Duke, "Matroid, Unimodularity, and Multicommodity Transportation Problem", Presented at ORSA/TIMS conference, Chicago, April 1975.

18. M. L. Fisher, "A Dual Algorithm for the One Machine Scheduling Problem", Technical Report 243, Operations Research Dept., Cornell University (1974).

19. M. L. Fisher and J. F. Shapiro, "Constructive Duality for Discrete Optimization", SIAM J. of A.M., 27, 1, 31-52 (1974).

20. H. Gabow, "An Efficient Implementation of Edmond's Algorithm for Maximum Matching on Graphs", to appear in JACM.

21. H. Gabow, F. Glover, and D. Klingman, "A Note on Exchanges in Matroid Bases", Center for Cybernetic Studies, Austin, Texas 78712 (1974).

22. Gavish and P. Schweitzer, "An Algorithm for Combining Truck Trips", Trans. Sci., 8, 1, 13-24 (1974).

23. A. M. Geoffrion and G. W. Graves, "Multicommodity Distribution System Design by Benders Decomposition", Man. Sci., 20, 5, 822-844 (1974).

24. F. Glover, "Compact L. P. Bases for a Class of I. P. Problems", NSRS 75-18, University of Colorado (1975).

25. F. Glover and D. Klingman, "Real World Applications of Network Related Problems and Breakthroughs in Solving Them Efficiently", ACM Transactions on Mathematical Software, 1, 1, 47-55 (1975).

26. F. Glover, D. Klingman, and G. T. Ross, "Finding Equivalent Transportation Formulations for Constrained Transportation Problems", Naval Research Logistics Quarterly 21, 2, 247-253 (1974)

27. F. Glover, D. Karney, and D. Klingman, "Implementation and Computational Study on Start Procedures and Basic Change Criteria for a Primal Network Code", Networks, 4, 3, 191-212 (1974).

28. F. Glover and D. Klingman, "A Note on Admissible Exchanges in Spanning Trees", MSRS 74-3, University of Colorado (1974).

29. F. Glover and D. Klingman, "Finding Minimum Spanning Trees with a Fixed Number of Lines at a Node", to appear in Opns. Res.

30. F. Glover, D. Karney, and D. Klingman, "An Extended Goaling Programming Formulation of Multi-Attribute Manpower Assignment Problems and Implementation of a Computer Assisted Policy Evaluation System", Technical Report, Navy Personnel Research and Development Laboratory, San Diego, California (1974).

31. F. Glover, D. Karney, and D. Klingman, "Augmented Predecessor Index Method for Location Stepping Stone Paths and Assigning Dual Prices in Distribution Problems", 6, 1, 171-181 (1972).

32. F. Glover, D. Karney, D. Klingman, and A. Napier, "A Computational Study on Start Procedures, Basis Change Criteria, and Solution Algorithms for Transportation Problems", Man. Sci., 20, 5, 793-819 (1974).

33. F. Glover and D. Klingman, "On the Equivalence of Some Generalized Network Problems to Pure Network Problems", Mathematical Programming, 4, 3, 369-378 (1973).

34. F. Glover, D. Klingman, and J. Stutz, "Extensions of the Augmented Predecessor Index Method to Generalized Network Problems", Trans. Sci., 7, 4, 377-384 (1973).

35. F. Glover, D. Klingman, and J. Stutz, "Implementation and Computational Study of a Generalized Network Code", 44th National Meeting of ORSA, San Diego, California (1973).

36. F. Glover and D. Klingman, "A Note on Computational Simplifications in Solving Generalized Transportation Problems", Trans. Sci., 7, 351-361 (1973).

37. F. Glover and D. Klingman, "Network Formulations of Cash Flow Problems", Research Report to appear Center for Cybernetic Studies, University of Texas, Austin, TX 78712 (1975).

38. F. Glover, D. Karney, D. Klingman, and R. Russell, "Solving Singularly Constrained Transshipment Problems, Research Report CCS 212, Center for Cybernetic Studies, University of Texas, Austin, Texas 78712 (1974).

39. F. Glover, D. Klingman, and J. Stutz, "Augmented Threaded Index Method for Network Optimization", INFOR, 12, 3, 293-298 (1974).

40. F. Glover and J. Mulvey, "Equivalence of the Zero-One Integer Program to Discrete Generalized and Pure Networks", MSRS 75-19, University of Colorado (1975).

41. F. Glover and D. Klingman, "Improved Labeling of L. P. Bases in Networks", Research Report CS 218, Center for Cybernetic Studies, University of Texas, Austin, Texas (1974)

42. F. Glover, D. Klingman, and J. Stutz, "Implementation and Testing on Integer Generalized Network Codes", Presented at Joint ORSA/TIMS Conference, San Juan, Puerto Rico (1974).

43. F. Glover and D. Klingman, "Double-pricing Dual and Feasible Start Algorithms for the Capacitated Transportation (distribution) Problems", University of Texas, Austin, Texas (1970).

44. F. Glover and D. Klingman, "Network Applications in Industry and Government", MSRS 75-20, University of Colorado, May 1975.

45. M. H. Glover, "A Network Model for the Distribution of Aggregates", to be presented at ORSA/TIMS conference, Las Vegas, Nov. 1975.

46. P. A. Jensen and G. Bhaumik, "A Flow Augmentation Approach to the Network with Gains Problems", Dept. of Mechanical Eng., University of Texas, 1974.

47. G. W. Graves and R. D. McBride, "The Use of Inherent Triangularity in Network Problems", presented at 44th National Meeting of ORSA, San Diego, November 1973.

48. W. W. Hogan, "Single Commodity Network Problems with Resource Constraints", Discussion Paper, U. S. Air Force Academy (1973).

49. E. Johnson, "Networks and Basic Solutions", Opns. Res. 14, 89-95 (1966).

50. D. Karney and D. Klingman, "Implementation and Computational Study on In-Core Out-of-Core Primal Network Code", to appear in Opns. Res.

51. J. Kennington, "The Fixed-Charge Transportation Problem", Technical Report CP 73-33, S.M.U.

52. D. Klingman and R. Russell, "On Solving Constrained Transportation Problems", Opns. Res. 1, 91-107 (1975).

53. D. Klingman, A. Napier, and G. Ross, "A Computational Study on the Effects of Problem Dimensions on Solution Time for Transportation Problems", to appear in JACM.

54. D. Klingman, A. Napier, and J. Stutz, "NETGEN-A Program for Generating Large Scale (Un) Capacitated Assignment, Transportation, and Minimum Cost Flow Network Problems", Man. Sci., 20, 5, 814-822 (1974).

55. D. Klingman and G. T. Ross, "Finding Equivalent Network Formulation for Constrained Network Problems", Research Report CCS 108, Center for Cybernetic Studies, University of Texas, Austin, Texas 78712 (1973), to appear in Man. Sci.

56. D. Klingman, P. Randolph, and S. Fuller, "A Cotton-Pickin Cotton Ginning Problem", to appear in Opns. Res.

57. R. W. Langley, "Continuous and Integer Generalized Flow Problems", Ph.D. Dissertation, Georgia Tech. (1973).

58. E. Lawler, "Combinatorial Optimization", Holt, Rhinehart, and Winston, to appear.

59. L. J. Leblanc and E. K. Morlok, "An Analysis and Comparison of Behavioral Assumptions in Traffic Assignment", International Symposium of Traffic Equilibrium Methods, Montreal, Quebec, November 1974.

60. L. J. Leblanc, "Mathematical Programming Algorithms for Large Scale Network Equilibrium and Network Design Problems", Ph.D. Dissertation, Dept. of Industrial Engineering and Management Sciences, Northwestern University (1973)

61. J. F. Maurras, "Optimization of the Flow Through Networks with Gains", Mathematical Programming, 4, 2, 135-145 (1972).

62. R. D. McBride, "Factorization in Large-Scale Linear Programming", Ph.D. Dissertation in Applied Mathematics, University of California, Los Angeles, June 1973.

63. J. Mulvey, "Column Weighting Factors and Other Enhancements to the Augmented Threaded Index Method for Network Optimization", Joint ORSA/TIMS Conference, San Juan, Puerto Rico (1974).

64. J.-M. Pla, "An Out-of-Kilter Algorithm for Solving Minimum Cost Potential Problems", Mathematical Programming 1, 275-290 (1971).

65. M. R. Rao and C. O. Fong, "Accelerated Labeling Algorithms for the Maximal Flow Problem with Applications to Transportation and Assignment Problems", W.P. No. 7222, University of Rochester, N.Y. (1973).

66. G. T. Ross and R. Soland, "A Branch and Bound Algorithm for the Generalized Assignment Problem", Mathematical Programming, 8, 1, 91-104 (1975).

67. J. F. Shapiro, "A Note on the Primal-Dual and Out-of-Kilter Algorithms for Network Optimization Problems", MIT, March 1975.

68. SIA Depot Location, "Poligami and Multicom, brochure, circa 1971.

69. R. M. Soland, "Optimal Facility Location with Concave Costs", Opns. Res., 22, 2, 373-382 (1974).

70. V. Srinivasan and G. L. Thompson, "Benefit-Cost Analysis of Coding Techniques for the Primal Transportation Algorithm", JACM, 20, 194-213 (1973).

71. V. Srinivasan and G. L. Thompson, "Accelerated Algorithms for Labeling and Relabeling of Trees with Application for Distribution Problems", JACM, 19, 4, 712-726 (1972).

72. "UKILT-1100 Programmer Reference Manual", UNIVAC, Data Processing and Mathematical Programming Division, Roseville, Minnesota.

73. H. P. Williams, "Experiments in the Formulation of Integer Programming Problems", Mathematical Programming, Study 2, 180-197 (1974).